

Amendments to the Claims

Please amend claims to be as follows.

1. (currently amended) A method ~~to dynamically generate an inline transformation order for call sites independent of an inline analysis order, wherein the method uses an inline affinity graph and an edge dependence graph.~~ of cross-file inlining during a compilation of a program, the method comprising determining which files to open and close based on affinity weightings between the files, wherein the affinity weightings depend on a number of potential inlines between the files.
2. (currently amended) A method of compiling a computer program from a plurality of files of source code, the method comprising:
 - an inline analysis to determine which call sites in the plurality of files to inline; and
 - an inline transformation to perform said inlining within currently opened files,~~wherein the inline transformation includes~~ including determining which files to open and close in dependence on ~~an affinity weighting~~ affinity weightings between the files.
3. (currently amended) The method of claim 2, wherein affinity weightings are representable by an inline affinity graph whose nodes correspond to files and whose edges correspond to potential inlines across corresponding files.
4. (currently amended) The method of claim 3, wherein the affinity weightings between files depend at least upon the number of potential inlines between the files.

5. (original) The method of claim 4, further comprising:
dynamically updating the inline affinity graph after inlinings within currently opened files are done.
6. (original) The method of claim 3, wherein an inline dependence for a call site is maintained including information as to a set of call sites that the call site depends upon.
7. (original) The method of claim 6, wherein inline dependencies are representable by an inline dependence graph.
8. (original) The method of claim 7, further comprising:
dynamically updating the inline dependence graph after inlinings within currently opened files are done.
9. (currently amended) ~~An apparatus to dynamically generate an inline transformation order for call sites independent of an inline analysis order, wherein the apparatus uses an inline affinity graph and an edge dependence graph. for compiling a program utilizing cross-file inlining, the apparatus comprising: a processor for executing instructions; a memory system for storing said instructions and data; and processor-executable instructions which determines which files to open and close based on affinity weightings between the files, wherein the affinity weightings depend on a number of potential inlines between the files.~~
10. (currently amended) An apparatus for compiling a computer program from a plurality of files of source code, the apparatus comprising:
a processor for executing instructions;
a memory system for storing said instructions and data;
processor-executable instructions for an analyzer module, the analyzer module being
configured to determine which call sites in the plurality of files to inline; and

processor-executable instructions for a transformer module, the transformer being configured to perform said inlining within currently opened files, wherein the transformer determines-including determining which files to open and close in dependence on an affinity weighting between the files.

11. (original) The apparatus of claim 10, wherein affinity weightings are representable by an inline affinity graph whose nodes correspond to files and whose edges correspond to affinity weightings between the files.
12. (original) The apparatus of claim 11, wherein the affinity weightings between files depend at least upon the number of inlines between the files.
13. (original) The apparatus of claim 10, wherein the transformer is further configured to dynamically update the inline affinity graph after inlinings within currently opened files are done.
14. (original) The apparatus of claim 11, wherein an inline dependence for a call site is maintained including information as to a set of call sites that the call site depends upon.
15. (original) The apparatus of claim 12, wherein inline dependencies are representable by an inline dependence graph.
16. (original) The apparatus of claim 13, wherein the transformer is further configured to dynamically update the inline dependence graph after inlinings within currently opened files are done.

17. (original) A computer program product comprising a computer-usable medium having computer-readable code embodied therein, the computer program product being compiled from a plurality of files of source code using an inline transformer which performs function inlining within currently opened files and determines which files to open and close in dependence on an affinity weighting between the files.